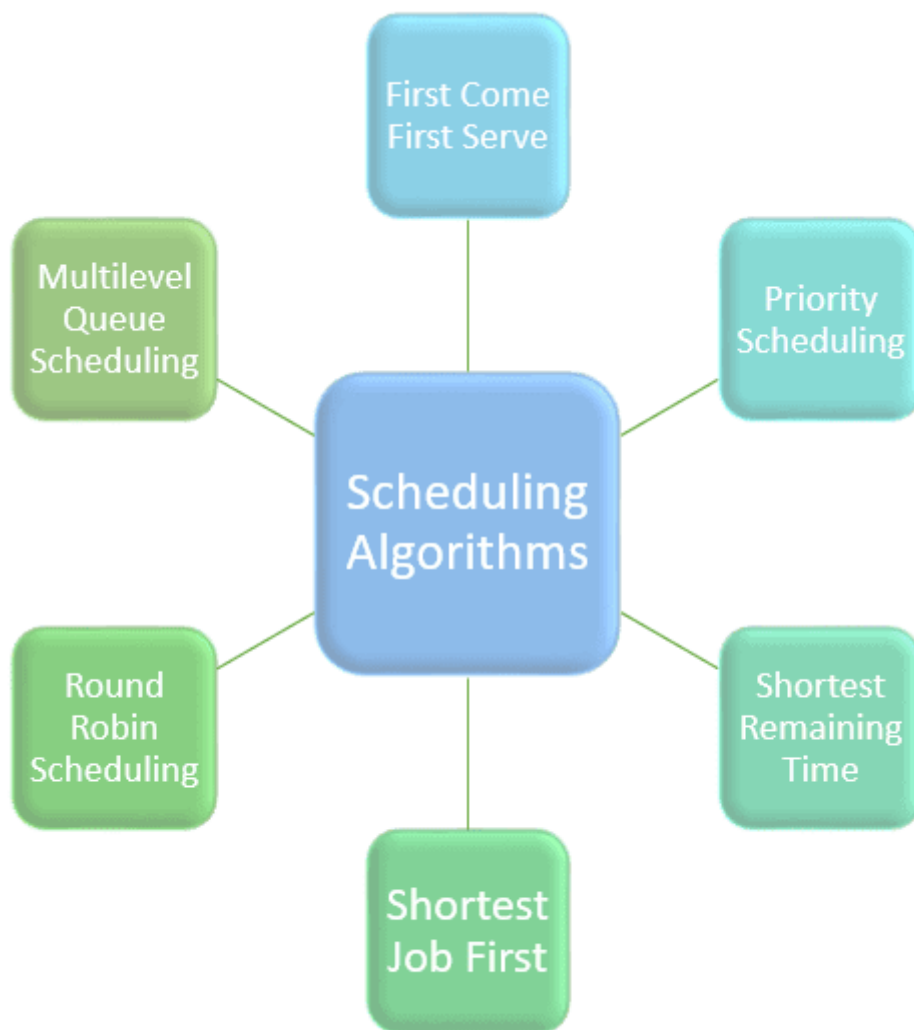


Λειτουργικά Συστήματα

Εξαμηνιαία Εργασία 2020-2021



Αναγνώστου Πανταζής 57497

Καρυπίδης Ευστάθιος 57556

Εισαγωγή

Σκοπός της εργασίας ήταν η προσομοίωση ενός λειτουργικού συστήματος μέσω της γλώσσας προγραμματισμού C. Ειδικότερα κληθήκαμε να υλοποιήσουμε και να συγκρίνουμε την πρακτική συμπεριφορά τεσσάρων από τους πιο δημοφιλείς αλγόριθμους χρονοδρομολόγησης/χρονοπρογραμματισμού διεργασιών. Οι αλγόριθμοι αυτοί είναι ο FCFS(First Come First Serve), ο LRU(Least Recently Used), ο Round Robin και ο Preemptive Priority. Αφού γίνει θεωρητική ανάλυση του προβλήματος και των επιμέρους αλγορίθμων, στην συνέχεια θα παρουσιαστεί ο κώδικας και η δομή του καθώς και τα παραδείγματα για τον έλεγχο του προγράμματος.

Μέρος Α: Θεωρητική Ανάλυση-Εξοικείωση

Πολύ συχνά στους σύγχρονους υπολογιστές συμβαίνει πολλές διεργασίες να ανταγωνίζονται για τον έλεγχο της CPU. Η κατάσταση αυτή εμφανίζεται όταν δύο ή περισσότερες διεργασίες είναι έτοιμες να εκτελεστούν. Στην περίπτωση που είναι διαθέσιμη μία μόνο CPU τότε πρέπει να ληφθεί μία απόφαση για το ποιά θα εκτελεστεί. Αυτό το αναλαμβάνει ένα τμήμα του λειτουργικού συστήματος το οποίο ονομάζεται χρονοπρογραμματιστής (scheduler) κάνοντας χρήση ενός αλγορίθμου που ονομάζεται αλγόριθμος χρονοπρογραμματισμού(scheduling algorithm). Συνοπτικά κάποιιοι από τους λόγους για τις οποίες χρησιμοποιούνται αλγόριθμοι χρονοπρογραμματισμού είναι οι εξής:

- Βέλτιστη αξιοποίηση της CPU
- Δίκαιη κατανομή της CPU
- Μέγιστη διεκπεραιωτική ικανότητα(throughput)
- Ελάχιστο χρόνο διεκπεραίωσης, ελάχιστο χρόνο αναμονής, ελάχιστο χρόνο απόκρισης

Στην εκφώνηση δίνεται ο ακόλουθος πίνακας με ένα σύνολο διεργασιών καθώς και διάφορα χαρακτηριστικά τους. Ειδικότερα για κάθε διεργασία δίνεται (α) η μονάδα χρόνου άφιξής της, (β) ο χρόνος απασχόλησης της Κεντρικής Μονάδας Επεξεργασίας (σε μονάδες χρόνου) τον οποίο χρειάζεται η διαδικασία, για να ολοκληρωθεί και (γ) η προτεραιότητά της (όσο μεγαλύτερος είναι ο αντίστοιχος αριθμός για κάθε διαδικασία, τόσο μεγαλύτερη είναι η προτεραιότητά της)

Διαδικασία(Process)	Χρόνος Άφιξης(Arrival Time)	Χρόνος Ολοκλήρωσης(Burst Time)	Προτεραιότητα(Priority)
P1	0	5	3
P2	1	6	5
P3	3	2	2
P4	9	4	1
P5	12	3	4

Με βάση το παραπάνω σύνολο διαδικασιών και για καθέναν από τους τέσσερις προαναφερθέντες αλγόριθμους χρονοδρομολόγησης, ζητείται να υλοποιηθούν ένας πίνακας της μορφής "Gantt chart" οποίος αποτελείται ουσιαστικά από δύο γραμμές, στην πρώτη από τις οποίες θα τυπώνονται οι διαδοχικές μονάδες χρόνου, ενώ στη δεύτερη θα τυπώνεται ο κωδικός αριθμός της διαδικασίας που χρησιμοποιεί την Κ.Μ.Ε. σε κάθε μονάδα χρόνου και ένα πίνακα μετρήσεων ο οποίος παρέχει (α) το χρόνο αναμονής, (β) το χρόνο διεκπεραίωσης, (γ) τον αριθμό εναλλαγών για κάθε διαδικασία, αφού ολοκληρωθεί η εκτέλεση όλων των διαδικασιών. Αρχικά ζητούνταν και ο χρόνος απόκρισης αλλά σε διάλεξη αναφέρθηκε να μην υπολογιστεί.

Διαμοιρασμός Φόρτου Εργασίας

Αναγνώστου Πανταζής :

- Αλγόριθμοι First Come First Serve, Round Robin
- Δομή Process, Gantt
- Συγγραφή Report
- Ατομικά Παραδείγματα

Καρυπίδης Ευστάθιος :

- Αλγόριθμοι Priority based preemptive, Shortest Remaining Time First
- Δομή ReadyQueue, VirtualCPU
- Συγγραφή Report
- Ατομικά Παραδείγματα

Χρήσιμοι ορισμοί

Χρόνος Αφίξης(Arrival time) = Η στιγμή όπου η διεργασία μπαίνει σε ready state και είναι έτοιμη να εκτελεστεί.

Χρόνος/Διάρκεια Ολοκλήρωσης(Burst time) = Ο χρόνος που απαιτείται από τη CPU για την εκτέλεση.

Χρόνος Διεκπεραίωσης(Turn Around Time-TAT) = Ο χρόνος διεκπεραίωσης είναι μέσος(στατιστικά) χρόνος που μεσολαβεί μεταξύ της υποβολής(κατάσταση ready) και της ολοκλήρωσης μιας διεργασίας. Μπορεί να υπολογιστεί ως εξής. **TAT = Exit/Completion time - Arrival time= (Burst time + Waiting time)**

Χρόνος Αναμονής(Waiting time-WT) = Ο χρόνος τον οποίο μία διαδικασία βρίσκεται στην ready queue. Υπολογίζεται ως εξής: **WT = TAT - Burst time)**

Μη προεκτοπιστικοί αλγόριθμοι(non-preemptive) Σε κάθε διακοπή ρολογιού επιλέγουν μια διεργασία η οποία θα εκτελεστεί μέχρι να επιστρέψει εθελονικά τη CPU(η μέχρι να εμφανιστεί κάποια διακοπή κλπ).

Προεκτοπιστικοί αλγόριθμοι(preemptive) Σε καθε διακοπή ρολογιού επιλέγεται μια διεργασία η οποία θα εκτελεστεί και συνεπώς μια εργασία μπορεί να διακοπεί και ο επεξεργαστής να δοθεί σε μια νέα διεργασία.

FCFS(First Come First Serve)

Ο **First Come First Serve (FCFS)** είναι από τους απλούστερους αλγόριθμους και όπου υποδηλώνει το όνομά του, η πρώτη διεργασία που φθάνει (στην ready queue) "καταλαμβάνει" την CPU. Οι διεργασίες εξυπηρετούνται ανάλογα με το χρόνο άφιξης τους πράγμα που επιτυγχάνεται μέσω μιας ουράς FIFO. Παρά την ευκολία κατανόησης και υλοποίησης ο FCFS δεν είναι ο ιδανικότερος και πιο αποδοτικός αλγόριθμος για διαμοιρασμό της CPU. Ειδικότερα μπορούν να εμφανιστούν εύκολα μεγάλοι χρόνοι αναμονής στην ready queue (να εμφανιστεί δηλαδή conroy effect) καθώς και το φαινόμενο όπου "σύντομες" διεργασίες να περιμένουν στην ουρά ώστε να τελειώσουν διεργασίες με μεγάλο χρόνο ολοκλήρωσης. Τέλος, όπως είναι εύκολα κατανοητό είναι ένας μη προεκτοπιστικός αλγόριθμος.

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P1	P1	P1	P1	P1	P2	P2	P2	P2	P2	P2	P3	P3	P4	P4	P4	P4	P5	P5	P5

	P1	P2	P3	P4	P5
Χρόνος Διεκπεραίωσης	5-0=5	11-1=10	13-3=10	17-9=8	20-12=8
Χρόνος Αναμονής	5-5=0	10-6=4	10-2=8	8-4=4	8-3=5
Αριθμός εναλλαγών	0	0	0	0	0

Priority based preemptive (PP)

Ο **Priority Scheduling** είναι ένας αλγόριθμος χρονοπρογραμματισμού με βάση την προτεραιότητα, δηλαδή εκτελούνται πρώτα οι διεργασίες όπου έχουν μεγαλύτερη προτεραιότητα με αποτέλεσμα αυτές να μην χρειάζεται να περιμένουν πολύ στην ready queue. Ο αλγόριθμος είναι προεκτοπιστικός και είναι κατάλληλος όταν υπάρχει ανάγκη για καλύτερη "παροχή υπηρεσιών", με δεδομένο ότι η αναγκαιότητα-προτεραιότητα των διαδικασιών είναι γνωστή. Ωστόσο, σε αυτήν την περίπτωση εάν οι διεργασίες υψηλής προτεραιότητας καταλαμβάνουν μεγάλο χρόνο στη CPU τότε οι διεργασίες χαμηλής προτεραιότητας συνεχώς αναβάλλονται και συνεπώς εύκολα μπορεί να εμφανιστούν φαινόμενα όπως το starvation.

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P1	P2	P2	P2	P2	P2	P2	P1	P1	P1	P1	P3	P5	P5	P5	P3	P4	P4	P4	P4

	P1	P2	P3	P4	P5
Χρόνος Διεκπεραίωσης	11-0=11	7-1=6	16-3=13	20-9=11	15-12=3
Χρόνος Αναμονής	(7-1)=6	0	(11-3)+(15-12)=8+3=11	(16-9)=7	0
Αριθμός εναλλαγών	1	0	1	0	0

Shortest Remaining Time First (SRTF)

Ο **SRTF** αυτός είναι η "προεκδοπιστική" εκδοχή του αλγορίθμου **SJF(Sortest Job First)**. Όπως υποδηλώνει και το όνομά του οι διεργασίες με το μικρότερο χρόνο ολοκλήρωσης εκτελούνται πρώτες και μία διεργασία μπορεί να αντικατασταθεί εάν μία άλλη κατά την άφιξή της έχει μικρότερο χρόνο ολοκλήρωσης. Η απόδοση του συγκεκριμένου αλγορίθμου είναι πολύ καλή έως ιδανική(μικρός μέσος χρόνος αναμονής και διεκπεραίωσης). Ωστόσο, σε περισσότερες πρακτικές εφαρμογές είναι δύσκολο να είναι γνωστός απο πριν ο χρόνος ολοκλήρωσης. Ακόμη, μπορούν και εδώ να εμφανιστούν και εδώ φαινόμενα μεγάλης αναμονής και ειδικότερα για διεργασίες με μεγάλους χρόνους εκτέλεσης. Τέλος επιλέξαμε ότι σε περίπτωση ίσων χρόνων ολοκλήρωσης να εκτελείται η διεργασία με μεγαλύτερο priority.

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P1	P1	P1	P1	P1	P3	P3	P2	P2	P2	P2	P2	P2	P5	P5	P5	P4	P4	P4	P4

	P1	P2	P3	P4	P5
Χρόνος Διεκπεραίωσης	5-0=5	13-1=12	7-3=4	20-9=11	16-12=4
Χρόνος Αναμονής	0	(7-1) = 6	(5-3)=2	(16-9)=7	(13-12)= 1
Αριθμός εναλλαγών	0	0	0	0	0

Round Robin (RR)

Ο αλγόριθμος **Round Robin** είναι απο τους πιο γνωστούς και παλιούς αλγορίθμους που μπορούν να υλοποιηθεί στα περισσότερα λειτουργικά συστήματα. Είναι προεκδοπιστικός αλγόριθμος και δίνει έμφαση στο διαμοιρασμό του χρόνου. Οι διεργασίες εκτελούνται με κυκλικό τρόπο για ένα συγκεκριμένο και ίσο χρονικό διάστημα η κάθε μία, το οποίο ονομάζεται κβάντο χρόνου. Εάν μια διεργασία ολοκληρωθεί σε αυτό το κβάντο τότε τερματίζεται αλλιώς επιστρέφει στην ready queue και περιμένει την επόμενη "επαναφορά/περιστροφή" να ολοκληρωθεί. Τα βασικά πλεονεκτήματα του συγκεκριμένου αλγορίθμου είναι ότι όλες οι διεργασίες διαμοιράζονται δίκαια την CPU, δεν εμφανίζονται φαινόμενα όπως το starvation η το connoy, δεν απαιτείται γνώση του χρόνου ολοκλήρωσης ή της προτεραιότητας και εμφανίζει καλό μέσο χρόνο αναμονής. Αντίθετα κάποια από τα μειονεκτήματα του αλγορίθμου είναι ότι δεν δίνεται κάποια προτεραιότητα σε "σημαντικές" διεργασίες, είναι δύσκολο να επιλεχθεί ένα "σωστό" κβάντου χρόνου καθώς επίσης η απόδοση του αλγορίθμου εξαρτάται πολύ απο την επιλογή του κβάντου χρόνου. Συνοπτικά, ένα μικρό κβάντο χρόνου μπορεί να οδηγήσει σε μεγαλύτερο overhead εναλλαγής σε ένα πραγματικό σύστημα ενώ ένα μεγάλο κβάντο χρόνου οδηγεί στη αύξηση του μέσου χρόνου αναμονής. Στο παράδειγμα που δώθηκε, ως κβάντο χρόνου ορίστηκαν οι 20 μονάδες(κατόπιν υπόδειξης) με αποτέλεσμα τα αποτελέσματα να είναι όμοια με τον FCFS.

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P1	P1	P1	P1	P1	P2	P2	P2	P2	P2	P2	P3	P3	P4	P4	P4	P4	P5	P5	P5

	P1	P2	P3	P4	P5
Χρόνος Διεκπεραίωσης	5-0=5	11-1=10	16-3=13	17-9=8	20-12=8
Χρόνος Αναμονής	5-5=0	10-6=4	10-2=8	8-4 = 4	8-3=5
Αριθμός εναλλαγών	0	0	0	0	0

Μέρος Β: Υλοποίηση

Όλοι οι αλγόριθμοι υλοποιήθηκαν σε ξεχωριστά header files ώστε να υπάρχει ευκολία στην κατανόηση τους. Πέρα των αλγορίθμων υπάρχει και ένα header file με κάποιες συναρτήσεις που χρησιμοποιούνται στο πρόγραμμα. Πιο συγκεκριμένα αυτή του sorting για τις διεργασίες που δίνονται από τον χρήστη. Ακόμη, στην λογική του κάθε αλγόριθμου δημιουργούνται οι δομές που περιγράψαμε παραπάνω και ακόμη γίνεται αρχικοποίηση τους στις μηδενικές τιμές που ορίσαμε οι ίδιοι.

Δομές:

Δομή Διεργασίας

Αρχικά, κατασκευάζουμε μία δομή της διεργασίας της οποίας θα επεξεργάζεται ο επεξεργαστής (**struct Process**). Αυτό το πρότυπο θα ακολουθούν όλες οι διεργασίες του συστήματος. Οι μεταβλητές που έχει η δομή της διεργασίας είναι οι παρακάτω:

- **name:** το οποίο είναι το πεδίο του ονόματος κάθε διεργασίας (μέχρι 3 χαρακτήρες).
- **arrive_time:** ο χρόνος που φτάνει κάθε διεργασία στον επεξεργαστή
- **burst_time:** ο χρόνος που θέλει μια διεργασία ώστε να ολοκληρωθεί
- **priority:** η προτεραιότητα σύμφωνα με την οποία πρέπει να διαχειριστεί η διεργασία

Σύμφωνα με αυτή τη δομή μπορούμε να φτιάξουμε όσες κενές διεργασίες θέλει ο χρήστης και στην συνέχεια να τις δώσει τις τιμές που ο ίδιος επιθυμεί.

Δομή ReadyQueue

Η δομή ReadyQueue είναι ουσιαστικά ένας πίνακας που αποτελείται από τις διεργασίες που έφτασαν στον επεξεργαστή αλλά δεν τρέχουν διότι ο επεξεργαστής είναι απασχολημένος. Ταξινομούνται μέσα στο πίνακα με βάση το ποιά διεργασία είναι αμέσως επόμενη να εκτελεστεί. Τα στοιχεία που την αποτελούν είναι τα παρακάτω:

- **processes_waiting:** Πίνακας τύπου Process όπου αποθηκεύονται οι διεργασίες που περιμένουν

Δομή VirtualCpu

Παρομοίως η δομή του VirtualCpu χρησιμοποιείται για την προσομοίωση μιας εικονικής CPU. Συνολικά στο πρόγραμμα, ανεξαρτήτου αλγορίθμου, πρέπει να υπάρχουν ισάριθμες εικονικές CPU με τις διαφορετικές διεργασίες που έχουμε. Τέλος, για την προσομοίωση της εικονικής CPU επιλέχθηκε να αναπαρασταθεί ως ένα σύνολο από τις διαδοχικές χρονικές στιγμές που τρέχουν οι αλγόριθμοι καθώς και στο ποιά κατάσταση βρίσκεται η εκάστοτε εικονική CPU εκείνη την στιγμή. Οι 4 καταστάσεις οι οποίες χρησιμοποιήθηκαν είναι οι παρακάτω:

- **N:** Η εικονική CPU δεν χρησιμοποιείται καθόλου και η αντίστοιχη διεργασία δεν έχει φτάσει ακόμα
- **W:** Η διεργασία έφτασε στη CPU ωστόσο η CPU είναι απασχολημένη άρα και περιμένει
- **U:** Η διεργασία χρησιμοποιείται από την CPU
- **S:** Η διεργασία σταμάτησε προσωρινά ώστε να εκτελεστεί μία άλλη

Δομή Gantt

Η δομή Gantt είναι υπεύθυνη για την προσομοίωση του Gantt chart καθώς και των χρόνων που ζητούνται αλλά και την παρουσίαση των δύο προηγούμενων δομών για κάθε χρονική στιγμή του CPU.

Αποτελείται από τις παρακάτω μεταβλητές:

- **algorithm_name:** κρατάει το όνομα του αλγορίθμου που χρησιμοποιήθηκε για την δημιουργία του Gantt
- **numberOfProcesses:** δηλώνει τον συνολικών διεργασιών που έχουμε
- **totalTime:** συνολικός χρόνος που χρειάζεται ώστε να εκτελεστούν όλες οι διεργασίες
- **states:** πίνακας τύπου Process ίσος με τον totalTime σε μέγεθος ώστε να υποδηλώνει την συνάρτηση που χρησιμοποιείτε κάθε στιγμή από το CPU.
- **waiting_times:** πίνακας με τον χρόνο που θα χρειαστεί κάθε διεργασία μέχρι να εκτελεστεί πρώτη φορά από το CPU
- **ta_times:** πίνακας με τους turnaround χρόνους κάθε διεργασίας
- **number_of_changes:** πίνακας με τον αριθμό των αλλαγών κάθε διεργασίας κατά την ολική εκτέλεση του προγράμματος
- **queues:** πίνακας τύπου ReadyQueue ώστε να έχουμε την κατάσταση της ReadyQueue για κάθε χρονική στιγμή
- **cpus:** πίνακας τύπου VirtualCpu ώστε να έχουμε την κατάσταση της διεργασίας κάθε διεργασίας μέσω της εικονικής CPU

Αλγόριθμοι

Για κάθε αλγόριθμο προσθέσαμε στο τέλος και τον υπολογισμό του μέσου χρόνου αναμονής και μέσου χρόνου διεκπεραίωσης.

FCFS

Αφού κάνουμε ταξινόμηση των διεργασιών με αύξοντα χρόνο άφιξης υλοποιούμε τον αλγόριθμο. Σύμφωνα με αυτόν, η K.M.E. παίρνει μία μία τις διεργασίες και τις εκτελεί για τον αντιστοίχο χρόνο ολοκλήρωσης τους. Κατά την διάρκεια της εκτέλεσης κάθε διεργασίας συμπληρώνεται και το χρονοδιάγραμμα του Gantt σύμφωνα με το ποιά διεργασία εκτελείται κάθε στιγμή. Τέλος, αφού ολοκληθούν οι διεργασίες υπολογίζονται όλοι οι χρόνοι που αναφέρθηκαν και παραπάνω. Παράλληλα με την διαδικασία εκτέλεσης μιας διεργασίας γίνεται έλεγχος για την άφιξη των υπόλοιπων ώστε να συμπληρώνονται το ReadyQueue (διεργασίες που περιμένουν) καθώς και τα virtualCPUs κάθε διεργασίας. Στην κορυφή του ReadyQueue βρίσκεται η διεργασία που θα εκτελεστεί ως επόμενη ενώ το virtualCPU κάθε διεργασίας αναπαριστά σε ποια κατάσταση βρίσκεται η ίδια για κάθε χρονική στιγμή. Να σημειωθεί πως σε αυτό τον αλγόριθμο μία διεργασία δεν μπορεί ποτέ να μπει σε κατάσταση sleeping (γράμμα S στα παραδείγματα) αφού αν επιλεγεί να εκτελεστεί από την K.M.E. δεν ξαναβγαίνει μέχρι την ολοκλήρωσή της.

Preemptive Priority

Για τον αλγόριθμο αυτό τρέχει ένα επαναλήπτικο κομμάτι όπου κάθε επανάληψη αναπαριστά την κάθε χρονική στιγμή της K.M.E. Κατά την διάρκεια της κάθε επανάληψης ο αλγόριθμος ελέγχει αρχικά αν έχει φτάσει μια νέα διεργασία και αν έχει μεγαλύτερη προτεραιότητα από αυτή της τωρινής (έχει οριστεί και μία "μηδενική" προτεραιότητα σε περίπτωση που δεν υπάρχει άλλη διεργασία στην K.M.E.). Αν ισχύουν τα προηγούμενα τότε επιλέγεται να εκτελεστεί αυτή η διεργασία για αυτή τη χρονική στιγμή, μειώνεται ο χρόνος ολοκλήρωσης της κατά ένα και στη συνέχεια μεταβαίνουμε στην επόμενη χρονική στιγμή. Όταν πλέον ο χρόνος ολοκλήρωσης όλων των διαδικασιών έχει φτάσει στο μηδέν τότε σημαίνει ότι ο αλγόριθμος έχει φτάσει στο τέλος του. Όπως και στον προηγούμενο αλγόριθμο άλλα και στους υπόλοιπους που θα παρουσιαστούν παρακάτω, παράλληλα με την εύρεση της τωρινής διεργασίας βρίσκουμε και ποιές διεργασίες έφτασαν αλλά δεν μπορούν να εξυπηρετηθούν αυτή τη χρονική στιγμή. Αυτές οι διεργασίες τοποθετούνται στη συνέχεια στην ReadyQueue πάλι με γνώμονα ποια θα εκτελεστεί στη συνέχεια. Ακόμη, συμπληρώνονται και τα virtualCPUs της κάθε διεργασίας ανάλογα την κατάσταση στην οποία βρίσκεται η ίδια. Σε αυτό τον αλγόριθμο εν αντίθεση του προηγούμενου μία διεργασία μπορεί να βρίσκεται σε κατάσταση sleeping αφού μπορεί να μπει στην K.M.E και στη συνέχεια μία άλλη διεργασία να πάρει την θέση της.

RoundRobin

Όπως εξηγήθηκε και παραπάνω ο αλγόριθμος της RoundRobin χρησιμοποιεί ένα κβάντο χρόνου ώστε να είναι διαμοιρασμένος σωστά ο χρόνος για κάθε διεργασία. Αυτό που κάνουμε στον αλγόριθμο είναι να εισάγουμε πάλι μια επαναληπτική διαδικασία η οποία ξεκινάει από την διεργασία που έφτασε πρώτη. Μετά ανάλογα τον υπολοιπόμενο χρόνο ολοκλήρωσης της μπαίνει σε μία απο τις δύο υλοποιήσεις. Αν ο υπολοιπόμενος χρόνος της είναι μικρότερος του κβάντου που δίνεται τότε η διεργασία υλοποιείται για τις υπόλοιπες μονάδες χρόνου της και στην συνέχεια μεταβαίνουμε στην επόμενη. Αν απο την άλλη είναι μεγαλύτερος του κβάντου τότε η διεργασία υλοποιείται μόνο για το κβάντο χρόνου και έπειτα μεταβαίνουμε στην επόμενη. Και στις 2 περιπτώσεις μειώνουμε τον χρόνο ολοκλήρωσης κατά πόσες φορές χρησιμοποιήθηκε η διεργασία ενώ λογικά προκύπτει πώς αν επιλεχθεί η 1η περίπτωση τότε και θα τελειώσει η συγκεκριμένη διεργασία. Η επιλογή της επόμενης διεργασίας που θα υλοποιηθεί γίνεται όπως πάντα με την κορυφή της ReadyQueue η οποία αναλογά και την περίπτωση που είμαστε συμπληρώνεται ανάλογα. Και σε αυτό τον αλγόριθμο έχουμε συμπλήρωση των virtualCPU ανάλογα την κατάσταση στην οποία βρίσκεται κάθε διεργασία κάθε χρονική στιγμή ενώ οι χρόνοι(waiting, turnaround κλπ κλπ) υπολογίζονται με το που η αντίστοιχη διεργασία ολοκληρωθεί. Τέλος, αφού ολοκληρωθούν όλες οι διεργασίες κάνουμε έξοδο από τον αλγόριθμο μας επιστρέφοντας το διάγραμμα Gantt πλήρως συμπληρωμένο.

Σημείωση: Κατά την συμπλήρωση του ReadyQueue κατά την διάρκεια μιας αλλαγής, σε περίπτωση που φτάσει μια νέα διεργασία και έχουμε και την διεργασία που υλοποιούταν πιο πριν στην κορυφή του ReadyQueue μπαίνει η νέα ως πιο δίκαιο σύστημα.

Shortest Remaining Time First (SRTF)

Όπως και στον Preemptive Priority τρέχει ένα επαναληπτικό κομμάτι όπου κάθε επανάληψη αναπαριστά την κάθε χρονική στιγμή της K.M.E. Κατα την διάρκεια των επαναλήψεων ελέγχει αν αρχικά φτάσει μία νέα διεργασία η οποία να έχει μικρότερο χρόνο ολοκλήρωσης. Σε περίπτωση ισότητας χρόνων ολοκλήρωσης επιλέγεται η διεργασία με την μεγαλύτερη προτεραιότητα. Με βάση αυτά επιλέγεται κάθε φορά η κατάλληλη διεργασία, μειώνεται ο χρόνος ολοκλήρωσης της κατά ένα και μεταβαίνουμε στην επόμενη χρονική στιγμή. Όταν όλες οι εργασίες ολοκληρωθούν(ελέγχονται αν οι χρόνοι ολοκλήρωσης γίνουν 0) τότε ο αλγόριθμος έχει τελειώσει. Αντιστοιχα με τους άλλους αλγορίθμους βρίσκονται τόσο οι διεργασίες οι οποίες έφτασαν αλλα δεν μπόρεσαν να εξυπηρετηθούν η εξυπηρετούνται και διακόπηκαν με αποτέλεσμα να μπουν στο ready queue. Ετσι, συμπληρώνονται και τα VirtualCPUS της κάθε διεργασίας ανάλογα με την κατάσταση που μπορεί να μπει στην K.M.E. Και εδώ οι διεργασίες μπορούν να μπουν και σε κατάσταση sleeping σε περίπτωση που καταφθάσει νέα διεργασία και να πάρει την θέση της παλιάς.

Screenshot Λειτουργίας - Παράδειγμα εκφώνησης

Αρχικά δίνουμε ως είσοδο τα δεδομένα:

```
Enter number of processes:5
Enter name of process:p1
Enter arrive,burst time and priority of p1: 0 5 3
Enter name of process:p2
Enter arrive,burst time and priority of p2: 1 6 5
Enter name of process:p3
Enter arrive,burst time and priority of p3: 3 2 2
Enter name of process:p4
Enter arrive,burst time and priority of p4: 9 4 1
Enter name of process:p5
Enter arrive,burst time and priority of p5: 12 3 4
Enter time quantum:20
```

Τα αποτελέσματα είναι τα εξής:

```
Time:0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20
FCFS:| p1 | p1 | p1 | p1 | p1 | p2 | p2 | p2 | p2 | p2 | p2 | p3 | p3 | p4 | p4 | p4 | p4 | p5 | p5 | p5 |
Ready Queue:
| :: | p2 | p2 | p2 | p2 | p3 | p3 | p3 | p3 | p3 | p3 | p3 | p4 | p4 | p5 | p5 | p5 | p5 | :: | :: | :: |
| :: | :: | :: | p3 | p3 | :: | :: | :: | :: | p4 | p4 | :: | p5 | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
Virtual CPUs:
Virtual CPU p1: | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU p2: | N | W | W | W | W | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N |
Virtual CPU p3: | N | N | N | N | W | W | W | W | W | W | W | U | U | N | N | N | N | N | N | N |
Virtual CPU p4: | N | N | N | N | N | N | N | N | N | N | W | W | W | W | U | U | U | U | N | N |
Virtual CPU p5: | N | N | N | N | N | N | N | N | N | N | N | N | N | N | W | W | W | W | U | U |

| Time | p1 | p2 | p3 | p4 | p5 |
| WT   | 0  | 4  | 8  | 4  | 5  |
| TA   | 5  | 10 | 10 | 8  | 8  |
| NC   | 0  | 0  | 0  | 0  | 0  |

Average Waiting Time: 4.20
Average Turnaround Time: 8.20

Time:0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20
preE:| p1 | p2 | p2 | p2 | p2 | p2 | p2 | p1 | p1 | p1 | p1 | p3 | p5 | p5 | p5 | p3 | p4 | p4 | p4 | p4 |
Ready Queue:
| :: | p1 | p1 | p1 | p1 | p1 | p1 | p3 | p3 | p3 | p3 | p4 | p3 | p3 | p3 | p4 | :: | :: | :: | :: |
| :: | :: | :: | p3 | p3 | p3 | p3 | :: | :: | p4 | p4 | :: | p4 | p4 | p4 | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
Virtual CPUs:
Virtual CPU p1: | U | S | S | S | S | S | S | S | U | U | U | U | N | N | N | N | N | N | N | N |
Virtual CPU p2: | N | U | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU p3: | N | N | N | W | W | W | W | W | W | W | W | U | S | S | S | U | N | N | N | N |
Virtual CPU p4: | N | N | N | N | N | N | N | N | N | N | W | W | W | W | W | W | U | U | U | U |
Virtual CPU p5: | N | N | N | N | N | N | N | N | N | N | N | N | N | N | U | U | U | N | N | N |

| Time | p1 | p2 | p3 | p4 | p5 |
| WT   | 6  | 0  | 11 | 7  | 0  |
| TA   | 11 | 6  | 13 | 11 | 3  |
| NC   | 1  | 0  | 1  | 0  | 0  |

Average Waiting Time: 4.80
Average Turnaround Time: 8.80
```

```

Time:0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20
RoRo:| p1 | p1 | p1 | p1 | p1 | p2 | p2 | p2 | p2 | p2 | p2 | p3 | p3 | p4 | p4 | p4 | p4 | p5 | p5 | p5 |
Ready Queue:
| :: | p2 | p2 | p2 | p2 | p3 | p3 | p3 | p3 | p3 | p3 | p3 | p3 | p4 | p4 | p4 | p5 | p5 | p5 | p5 | :: | :: | :: |
| :: | :: | :: | p3 | p3 | :: | :: | :: | :: | :: | p4 | p4 | :: | p5 | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
Virtual CPUs:
Virtual CPU p1: | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU p2: | N | W | W | W | W | U | U | U | U | U | U | U | U | U | N | N | N | N | N | N | N | N |
Virtual CPU p3: | N | N | N | W | W | W | W | W | W | W | W | W | W | U | U | N | N | N | N | N | N | N |
Virtual CPU p4: | N | N | N | N | N | N | N | N | N | N | N | W | W | W | W | U | U | U | U | N | N | N |
Virtual CPU p5: | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | W | W | W | W | W | U | U |

| Time | p1 | p2 | p3 | p4 | p5 |
| WT   | 0  | 4  | 8  | 4  | 5  |
| TA   | 5  | 10 | 10 | 8  | 8  |
| NC   | 0  | 0  | 0  | 0  | 0  |

Average Waiting Time: 4.20
Average Turnaround Time: 8.20

Time:0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20
SRTF:| p1 | p1 | p1 | p1 | p1 | p3 | p3 | p2 | p2 | p2 | p2 | p2 | p2 | p5 | p5 | p5 | p4 | p4 | p4 | p4 |
Ready Queue:
| :: | p2 | p2 | p3 | p3 | p2 | p2 | :: | :: | p4 | p4 | p4 | p5 | p4 | p4 | p4 | p4 | :: | :: | :: | :: |
| :: | :: | :: | p2 | p2 | :: | :: | :: | :: | :: | :: | :: | p4 | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
Virtual CPUs:
Virtual CPU p1: | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU p2: | N | W | W | W | W | W | W | U | U | U | U | U | U | U | N | N | N | N | N | N | N |
Virtual CPU p3: | N | N | N | W | W | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU p4: | N | N | N | N | N | N | N | N | N | N | W | W | W | W | W | W | W | U | U | U | U |
Virtual CPU p5: | N | N | N | N | N | N | N | N | N | N | N | N | N | N | W | U | U | U | N | N | N |

| Time | p1 | p2 | p3 | p4 | p5 |
| WT   | 0  | 6  | 2  | 7  | 1  |
| TA   | 5  | 12 | 4  | 11 | 4  |
| NC   | 0  | 0  | 0  | 0  | 0  |

Average Waiting Time: 3.20
Average Turnaround Time: 7.20

```

Προσωπικό Παράδειγμα

Σημείωση: Και στα 2 παραδείγματα που παρουσιάζονται επιλέχθηκε σαν κβάντο για τον RoundRobin ο αριθμός των 2 χρονικών μονάδων έναντι του 20 που δινόταν από την εκφώνηση διότι αλλιώς τα αποτελέσματα του θα ήταν ίδια με του FCFS.

Παράδειγμα 1^ο:

Έχουμε το παρακάτω παράδειγμα από διεργασίες που φτάνουν στην Κ.Μ.Ε.:

Διαδικασία (Process)	Χρόνος Άφιξης (Arrival Time)	Χρόνος Ολοκλήρωσης (Burst Time)	Προτεραιότητα (Priority)
P1	0	9	5
P2	1	4	3
P3	2	5	1
P4	3	7	2
P5	4	3	4

Τότε τα θεωρητικά αποτελέσματα είναι τα παρακάτω:

Αλγόριθμος	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
FCFS	P1	P1	P1	P1	P1	P1	P1	P1	P1	P2	P2	P2	P2	P3	P3	P3	P3	P3	P4	P4	P4	P4	P4	P4	P4	P5	P5	P5
preE	P1	P1	P1	P1	P1	P1	P1	P1	P1	P5	P5	P5	P2	P2	P2	P2	P4	P4	P4	P4	P4	P4	P4	P3	P3	P3	P3	P3
RoRo	P1	P1	P2	P2	P3	P3	P1	P1	P4	P4	P5	P5	P2	P2	P3	P3	P1	P1	P4	P4	P5	P3	P1	P1	P4	P4	P1	P4
SRTF	P1	P2	P2	P2	P2	P5	P5	P5	P3	P3	P3	P3	P3	P4	P4	P4	P4	P4	P4	P4	P1	P1	P1	P1	P1	P1	P1	P1

Ενώ οι θεωρητικοί χρόνοι είναι οι παρακάτω (η σειρά των χρόνων είναι FCFS - preE - RoRo - SRTF) και φαίνονται οι 2 χρόνοι που ζητούνται και ο αριθμός των εναλλαγών (3 πεδία για κάθε αλγόριθμο):

Αλγόριθμος		FCFS			preE			RoRo			SRTF	
Χρόνοι/ Διεργασία	Turnaround	Waiting	Number Of Changes	Turnaround	Waiting	Number Of Changes	Turnaround	Waiting	Number Of Changes	Turnaround	Waiting	Number Of Changes
P1	9-0=9	9-9=0	0	9-0=9	9-9=0	0	27-0=27	27-9=18	4	28-0=28	28-9=19	1
P2	13-1=12	12-4=8	0	16-1=15	15-4=11	0	14-1=13	13-4=9	1	5-1=4	4-4=0	0
P3	18-2=16	16-5=11	0	28-2=26	26-5=21	0	22-2=20	20-5=15	2	13-2=11	11-5=6	0
P4	25-3=22	22-7=15	0	23-3=20	20-7=13	0	28-3=25	25-7=18	3	20-3=17	17-7=10	0
P5	28-4=24	24-3=21	0	12-4=8	8-3=5	0	21-4=17	17-3=14	1	8-4=4	4-3=1	0

Μέσοι Χρόνοι	FCFS	preE	RoRo	SRTF
Average Waiting Time	55/5 = 11	50/5=10	74/5=14.8	36/5=7.2
Average Turnaround Time	83/5=16.6	78/5=15.6	102/5=20.4	64/5=12.8

Σχόλια: Όπως παρατηρούμε ο αλγόριθμος SRTF είναι ο πιο αποδοτικός (έχει τους λιγότερους μέσους χρόνους) ωστόσο δεν είναι και υλοποιήσιμος στην πραγματικότητα αφού δεν μπορούμε να γνωρίζουμε τον χρόνο ολοκλήρωσης των διεργασιών. Αμέσως μετά καλύτερος φαίνεται ο Priority Based ενώ μετά ο FCFS και τέλος ο RoundRobin. Ο RoundRobin παρόλο που είναι ο λιγότερο αποδοτικός, είναι ο πιο δίκαιος αφού γίνεται δίκαια κατανομή του χρόνου. Μεγάλο πλεονέκτημα ακόμα του αλγόριθμου αυτού είναι ότι δεν εμφανίζεται το convey effect όπως φαίνεται και στα screenshot παρακάτω. Ακόμη, όπως και αναφέρθηκε παραπάνω και φαίνεται και στα screenshot στη συνέχεια, στον αλγόριθμο FCFS παρατηρείται το convey effect όπου οι διεργασίες περιμένουν να έρθει η σειρά τους για να εκτελεστούν παραμένοντας στάσιμες μέχρι τότε στο ReadyQueue. Επίσης, και στο FCFS (P5) και στο preE (P3) παρατηρείται το φαινόμενο του starvation που περιγράψαμε και πάνω αφού οι 2 αυτές διεργασίες έχουν σχετικά μικρό χρόνο ολοκλήρωσης σε σχέση με τον χρόνο που περιμένουν. Τέλος, ο αριθμός των εναλλαγών είναι σε φυσιολογικά επίπεδα με τις περισσότερες αλλαγές να παρατηρούνται όπως είναι λογικό στον RoundRobin.

Και παρακάτω φαίνονται τα screenshots ότι τα αποτελέσματα μας ταιριάζουν:

```
Enter number of processes:5
Enter name of process:P1
Enter arrive,burst time and priority of P1: 0 9 5
Enter name of process:P2
Enter arrive,burst time and priority of P2: 1 4 3
Enter name of process:P3
Enter arrive,burst time and priority of P3: 2 5 1
Enter name of process:P4
Enter arrive,burst time and priority of P4: 3 7 2
Enter name of process:P5
Enter arrive,burst time and priority of P5: 4 3 4
Enter time quantum:2
```

```

Time:0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28
FCFS: | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 | P3 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P4 | P4 | P4 | P5 | P5 | P5 |

Ready Queue:
| :: | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | P5 | P5 | P5 | P5 |
| :: | :: | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | P4 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | P5 | P5 | P5 | P5 | P5 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:
Virtual CPU P1: | U | U | U | U | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P2: | N | W | W | W | W | W | W | W | W | W | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P3: | N | N | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P4: | N | N | N | N | W | W | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U | N | N | N | N |
Virtual CPU P5: | N | N | N | N | N | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U |

| Time | P1 | P2 | P3 | P4 | P5 |
| WT | 0 | 8 | 11 | 15 | 21 |
| TA | 9 | 12 | 16 | 22 | 24 |
| NC | 0 | 0 | 0 | 0 | 0 |

Average Waiting Time: 11.00
Average Turnaround Time: 16.60

Time:0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28
preF: | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P5 | P5 | P5 | P2 | P2 | P2 | P2 | P4 | P4 | P4 | P4 | P4 | P4 | P4 | P3 | P3 | P3 | P3 | P3 |
Ready Queue:
| :: | P2 | P2 | P2 | P5 | P5 | P5 | P5 | P5 | P2 | P2 | P2 | P2 | P4 | P4 | P4 | P4 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 |
| :: | :: | P3 | P4 | P2 | P2 | P2 | P2 | P4 | P4 | P4 | P3 | P3 | P3 | P3 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | P3 | P4 | P4 | P4 | P4 | P3 | P3 | P3 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | P3 | P3 | P3 | P3 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:
Virtual CPU P1: | U | U | U | U | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P2: | N | W | W | W | W | W | W | W | W | W | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P3: | N | N | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U | U | U |
Virtual CPU P4: | N | N | N | N | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U | U | U | N | N | N |
Virtual CPU P5: | N | N | N | N | N | W | W | W | W | W | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N |

| Time | P1 | P2 | P3 | P4 | P5 |
| WT | 0 | 11 | 21 | 13 | 5 |
| TA | 9 | 15 | 26 | 20 | 8 |
| NC | 0 | 0 | 0 | 0 | 0 |

Average Waiting Time: 10.00
Average Turnaround Time: 15.60

```

```

Time:0    1    2    3    4    5    6    7    8    9    10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28
Rofo:| P1 | P1 | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P5 | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P3 | P1 | P1 | P4 | P4 | P1 | P4 |

Ready Queue:

| :: | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P5 | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P5 | P3 | P1 | P1 | P4 | P4 | P1 | P1 | P4 | :: |
| :: | :: | :: | P1 | P1 | P4 | P4 | P5 | P5 | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P5 | P3 | P3 | P1 | P4 | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | P4 | P5 | P5 | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P5 | P3 | P3 | P1 | P1 | P4 | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P5 | P5 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:

Virtual CPU P1: | U | U | S | S | S | S | U | U | S | S | S | S | S | S | S | S | S | S | S | S | U | U | S | S | S | U | N | N |
Virtual CPU P2: | N | N | U | U | U | S | S | S | S | S | S | S | S | S | U | U | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P3: | N | N | N | W | W | U | U | S | S | S | S | S | S | S | S | S | S | S | S | S | S | U | N | N | N | N | N | N |
Virtual CPU P4: | N | N | N | N | W | W | W | W | W | W | U | U | S | S | S | S | S | S | S | S | S | S | S | S | U | U | S | U | N |
Virtual CPU P5: | N | N | N | N | N | W | W | W | W | W | W | W | U | U | S | S | S | S | S | S | S | S | U | S | N | N | N | N | N |

| Time | P1 | P2 | P3 | P4 | P5 |
| WT | 18 | 9 | 15 | 18 | 14 |
| TA | 27 | 13 | 20 | 25 | 17 |
| NC | 4 | 1 | 2 | 3 | 1 |

Average Waiting Time: 14.80
Average Turnaround Time: 20.40

Time:0    1    2    3    4    5    6    7    8    9    10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28
SRFT:| P1 | P2 | P2 | P2 | P5 | P5 | P5 | P3 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P4 | P4 | P4 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 |

Ready Queue:

| :: | P1 | P3 | P3 | P5 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | P1 | P1 | P4 | P4 | P4 | P4 | P1 | P1 | P1 | P1 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | P1 | P4 | P1 | P1 | P1 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |
| :: | :: | :: | :: | P1 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:

Virtual CPU P1: | U | S | S | S | S | S | S | S | S | S | S | S | S | S | S | S | S | S | S | S | U | U | U | U | U | U | U |
Virtual CPU P2: | N | N | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P3: | N | N | N | W | W | W | W | W | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P4: | N | N | N | N | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U | U | U | N | N | N | N | N |
Virtual CPU P5: | N | N | N | N | N | W | N | W | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |

| Time | P1 | P2 | P3 | P4 | P5 |
| WT | 19 | 0 | 6 | 10 | 1 |
| TA | 28 | 4 | 11 | 17 | 4 |
| NC | 1 | 0 | 0 | 0 | 0 |

Average Waiting Time: 7.20
Average Turnaround Time: 12.80

```

Παράδειγμα 2^ο:

Σε αυτό το παράδειγμα έχουμε τις παρακάτω διεργασίες:

Διαδικασία (Process)	Χρόνος Αφίξης (Arrival Time)	Χρόνος Ολοκλήρωσης (Burst Time)	Προτεραιότητα (Priority)
P1	0	5	3
P2	3	12	1
P3	4	11	2

και τα αποτελέσματα του διαγράμματος Gantt για όλους τους αλγορίθμους είναι τα παρακάτω:

Αλγόριθμος	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
FCFS	P1	P1	P1	P1	P1	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3
preE	P1	P1	P1	P1	P1	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2
RoRo	P1	P1	P1	P1	P2	P2	P3	P3	P1	P2	P2	P3	P3	P2	P2	P3	P2	P2	P3	P3	P2	P2	P3	P3	P2	P2	P2	P3
SRTF	P1	P1	P1	P1	P1	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2	P2

και οι αντίστοιχοι χρόνοι και αριθμοί εναλλαγών είναι:

Αλγόριθμος		FCFS			preE			RoRo			SRTF	
Χρόνοι/ Διεργασία	Turnaround	Waiting	Number Of Changes	Turnaround	Waiting	Number Of Changes	Turnaround	Waiting	Number Of Changes	Turnaround	Waiting	Number Of Changes
P1	5-0=5	5-5=0	0	5-0=5	5-5=0	0	9-0=9	9-5=4	1	5-0=5	5-5=0	0
P2	17-3=14	14-12=2	0	28-3=25	25-12=13	0	27-3=24	24-12=12	5	28-3=25	25-12=13	0
P3	28-4=24	24-11=13	0	16-4=12	12-11=1	0	28-4=24	24-11=13	5	16-4=12	12-11=1	0

Μέσοι Χρόνοι	FCFS	preE	RoRo	SRTF
Average Waiting Time	15/3=5	14/3=4.67	29/3=9.67	14/3=4.67
Average Turnaround Time	43/3=14.33	42/3=14	57/3=19	42/3=14

Σχόλια: Και σε αυτό το παράδειγμα παρατηρούμε ότι καλύτερος αλγόριθμος είναι ο SRTF με τον Priority Based να είναι το ίδιο αποδοτικός. Η αποδοτικότητα των αλγορίθμων είναι ιεραρχικά ίδια με πριν με τον RoundRobin πάλι να είναι ο λιγότερο αποδοτικός αλλά και ο πιο δίκαιος. Το convey effect παρατηρείται πάλι στον FCFS αλλά για ελάχιστο χρονικό διάστημα λόγω του μικρού αριθμού διεργασιών. Τέλος, παρατηρούμε starvation effect στον PriorityBased όπου η P2 χρειάζεται να περιμένει να ολοκληρωθεί η P3 για να ξεκινήσει λόγω της χαμηλής προτεραιότητας που έχει. Στο παράδειγμα αυτό ο αριθμός των εναλλαγών είναι διάφορος του μηδενός μόνο στον RoundRobin που αφού έχουμε και το κβάντο χρόνου είναι σωστό.

Και παρακάτω είναι τα screenshots λειτουργίας του παραδείγματος επαληθεύοντας τα παραπάνω θεωρητικά αποτελέσματα:

```

Enter number of processes:3
Enter name of process:P1
Enter arrive,burst time and priority of P1: 0 5 3
Enter name of process:P2
Enter arrive,burst time and priority of P2: 3 12 1
Enter name of process:P3
Enter arrive,burst time and priority of P3: 4 11 2
Enter time quantum:2

```

```

Time:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
FCFS:| P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 |

Ready Queue:
| :: | :: | :: | P2 | P2 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 |
| :: | :: | :: | :: | P3 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:
Virtual CPU P1: | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P2: | N | N | N | N | W | W | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | N | N | N | N | N | N |
Virtual CPU P3: | N | N | N | N | N | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U |

| Time | P1 | P2 | P3 |
| WT | 0 | 2 | 13 |
| TA | 5 | 14 | 24 |
| MC | 0 | 0 | 0 |

Average Waiting Time: 5.00
Average Turnaround Time: 14.33

Time:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
preE:| P1 | P1 | P1 | P1 | P1 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 |

Ready Queue:
| :: | :: | :: | P2 | P3 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 |
| :: | :: | :: | :: | P2 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:
Virtual CPU P1: | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P2: | N | N | N | N | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U |
Virtual CPU P3: | N | N | N | N | N | W | U | U | U | U | U | U | U | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N |

| Time | P1 | P2 | P3 |
| WT | 0 | 13 | 1 |
| TA | 5 | 25 | 12 |
| MC | 0 | 0 | 0 |

Average Waiting Time: 4.67
Average Turnaround Time: 14.00

```

```

Time:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
RR:| P1 | P1 | P1 | P1 | P1 | P2 | P2 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P2 | P2 | P2 | P2 | P2 | P2 | P3 | P3 | P3 | P3 | P3 |
| P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 |

Ready Queue:
| :: | :: | :: | P2 | P3 | P3 | P3 | P1 | P1 | P2 | P3 | P3 | P3 | P2 | P2 | P3 | P3 | P2 | P2 | P3 | P3 | P2 | P2 | P3 | P3 | P3 | P3 |
| :: | :: | :: | :: | P1 | P1 | P2 | P2 | P3 | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: | :: |

Virtual CPUs:
Virtual CPU P1: | U | U | U | U | U | S | S | S | S | S | U | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P2: | N | N | N | N | W | U | U | S | S | S | U | U | S | S | U | U | S | S | U | U | S | S | U | U | S | S | U | U |
Virtual CPU P3: | N | N | N | N | N | W | W | U | U | U | S | S | S | U | U | S | S | U | U | S | S | U | U | S | S | U | U | S | U |

| Time | P1 | P2 | P3 |
| WT | 4 | 12 | 13 |
| TA | 9 | 24 | 24 |
| MC | 1 | 5 | 5 |

Average Waiting Time: 9.67
Average Turnaround Time: 19.00

Time:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
SRIF:| P1 | P1 | P1 | P1 | P1 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P3 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 |

Ready Queue:
| :: | :: | :: | P2 | P3 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 |
| :: | :: | :: | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 |

Virtual CPUs:
Virtual CPU P1: | U | U | U | U | U | W | U | W | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N | N |
Virtual CPU P2: | N | N | N | N | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | U | U | U | U | U | U |
Virtual CPU P3: | N | N | N | N | N | W | U | U | U | U | U | U | U | U | U | U | U | U | U | N | N | N | N | N | N | N | N | N |

| Time | P1 | P2 | P3 |
| WT | 0 | 13 | 1 |
| TA | 5 | 25 | 12 |
| MC | 0 | 0 | 0 |

Average Waiting Time: 4.67
Average Turnaround Time: 14.00

```

Πηγές

1. <https://www.guru99.com/cpu-scheduling-algorithms.html#9>
2. <https://www.guru99.com/fcfs-scheduling.html>
3. <https://www.guru99.com/round-robin-scheduling-example.html>
4. <https://www.guru99.com/priority-scheduling-program.html>
5. <https://www.guru99.com/shortest-job-first-sjf-scheduling.html>
6. <https://www.javatpoint.com/os-scheduling-algorithms>
7. https://www.tutorialspoint.com/operating_system/os_process_scheduling_algorithms.htm
8. <https://afteracademy.com/blog/what-is-burst-arrival-exit-response-waiting-turnaround-time-and-throughput>
9. Tanenbaum, Andrew S_Bos, Herbert - Modern operating systems - Pearson Education (Us)_ Langara College (2016)
10. Διαφάνειες Μαθήματος: Διάλεξη 9-10 Χρονοπρογραμματισμός opsys5-scheduling